

PATIENT RULE INDUCTION METHOD ON LARGE DISK
RESIDENT DATA SETS AND PARALLELIZATION THEREOF

5 The present invention relates generally to data analysis involving the prediction of high values of some output variable based on a set of input variables. In particular, the invention relates to analysis of large, disk resident data sets using the Patient Rule Induction Method (PRIM).

BACKGROUND OF THE INVENTION

10 A generic problem in data analysis involves the prediction of high values of some output variable based on a set of input variables. For example, a hospital might want to determine from collected patient data what combinations of patient age and length of stay are associated with high cost hospital stays. Adding to the complexity of this data analysis problem is the fact that the variables of interest can be of different types, including:

15 continuous/numerical (ordered data, such as age, cost, length of stay);
categorical (non-ordered data, such as doctor); and
discrete (ordered, non-continuous data, such as procedure risk).

20 There are many prior art strategies for solving similar data analysis problems to the one described above, including the SLIQ classification algorithm (M. Mehta, R. Agrawal and J. Rissanen, *SLIQ: A Fast Scalable Classifier for Data Mining*, Proc. of the Fifth Int'l Conference on Extending Database Technology, Avignon, France, March 1996.), SPEC (*An Efficient, Scalable, Parallel Classifier for Data Mining* (1996) Anurag Srivastava, Vineet Singh, Eui-Hong (Sam) Han and Vipin Kumar, Technical Report 96-040, Department of

Computer Science, University of Minnesota, Minneapolis, 1996.), and the Patient Rule Induction Method (PRIM) (Friedman, J.H. and Fisher, N. I. *Bump Hunting in High-Dimensional Data* (Oct. 1997) (to appear in Statistics and Computation). In each of these strategies, the data to be analyzed is stored initially in a database of any type and the data analysis system copies the stored data onto disk using specific data structures that facilitate analysis of the data.

The SLIQ classification algorithm finds local groups of the input data that have a high output value. It works better with categorical data than with continuous data. That is, the SLIQ algorithm is more likely to find contiguous groups of multi-dimensional input data with high output values for categorical rather than continuous data. Because of this, the SLIQ algorithm is typically paired with linear regression analysis when continual data needs to be analyzed.

The PRIM algorithm, on which the present invention is based, specifically addresses the issue of determining regions in a multidimensional space of a set of input variables that correspond to high average for an output variable. Moreover, the regions generated by PRIM are easily interpretable as they are rectangles for two input variables or (more generally) hyper-rectangles for more than two input variables. (Note, for the remainder of this application, the terms “hyper-rectangles” and “boxes” are used interchangeably.) A comparison of the results given by the SLIQ and PRIM algorithms for the same input is now described with reference to Figure 1.

Referring to Figure 1, there is shown a plot where the input variables are *Length Of Stay* and *Age* for hospital patients and the output variable is *Charge* for the patient’s stay, where points that are filled-in are high value points and points that are not filled in are low value points. Given the goal of finding those regions of the *Length of Stay/Age* space with highest *Charge*, the PRIM algorithm would return the single hyper-rectangle 102, whereas the SLIQ algorithm would return the smaller, disjoint regions 104a, 104b. The PRIM hyper-rectangle region 102 description (e.g., $50 < \text{Age} < 70$ and $2 \leq \text{Stay Length} \leq 4$) can be used for prediction applications where the input variables are known and the output variable is

unknown. In particular, for a hospital application, the high charge (or high cost) patients may be used to reduce costs internally or to negotiate more profitable contracts with payors (or insurance companies) externally. This application is facilitated by the simpler output of PRIM as compared to that from the SLIQ algorithm. As the present invention is based on PRIM, this method is now described in greater detail.

The idea behind PRIM is to enclose all input data points within a single hyper-rectangle and then successively peel away contiguous, small strips of the hyper-rectangle enclosing low value points until a user-defined percentage of the total points remain inside the hyper-rectangle. These points are presumed to be predominantly high value points.

PRIM can be applied to a small data set that can be copied in its entirety into main memory (i.e., RAM accessible to executing programs), but is not able to handle larger data sets.

PRIM includes the following steps for continuous data:

- 1) copy the entire data set into main memory;
- 2) define a peeling fraction (α) which is the width of the contiguous, small strips of the hyper-rectangle. Each strip is formed by grabbing a given percentage of the initial points which is preferable between 1% and 5% of the points along the perimeter of the hyper-rectangle.
- 3) form the strips along edges of the hyper-rectangle by taking $\alpha\%$ of the points along the perimeter of the hyper-rectangle.
- 4) calculate an average value of the cost attribute for each strip of the hyper-rectangle. For each attribute, sort the data and find the average value of the cost attribute for $\alpha\%$ of the top points and $\alpha\%$ of the bottom points. This is repeated for each attribute.
- 5) throw away the strip with the lowest average cost value and return to step 3) - repeat until the number of points remaining enclosed by the hyper-rectangle equals a user-defined percentage of the initial points (the remaining points will be the user-defined percentage of points with the highest average output value).

The steps for discrete data are similar to the previously described steps, except, instead of multi-dimensional strips, histograms are used. For example, if one of the input

variables were *Doctor ID*, each strip would correspond to a histogram bin for a given *Doctor ID*. The way PRIM is implemented for small data sets is now described with reference to Figure 2.

5 Referring to Figure 2, there is shown an illustration of a data structure used by the prior art PRIM implementation for discrete attributes and a relational table of data. The relation table of data contains both continuous attributes and discrete attributes. To process the discrete attribute *Doctor ID*, a histogram is generated containing all distinct *Doctor IDs* and an expected cost. The expected cost is calculated by summing the cost for each record 10 with each *Doctor ID* and dividing the sum by a count of each record with the *Doctor ID*. As depicted in FIG 2, the *Doctor IDs* A, B, C and D are shown along with the expected cost. The *Doctor ID* A has the lowest expected cost (\$100.00) and is therefore chosen as the peeling discrete attribute.

15 Consequently, the PRIM calculation for discrete attributes is exactly as described above for continuous. However for discrete attributes, step 4) would include forming a histogram for each discrete attribute containing the distinct discrete attribute values and expected cost. Step 5) would include comparing the continuous attribute with the lowest average value to the discrete attribute with the lowest average value to determine the peeling attribute to be removed.

20 The PRIM implementation described above must sort the relation table of data for each attribute one at a time for each peeling step. In addition, the relational table of data must be queried to determine the expected cost for each discrete attribute for each peeling step. 25 These tasks require that the relational table of data be stored in main memory. Unfortunately, memory limitations of conventional systems prevent large disk resident data sets from being entirely sorted in main memory. Therefore, there is a need for an implementation of PRIM that is applicable to large, disk-resident data sets. There is also a need for an implementation of PRIM that can be parallelized for execution on parallel processors.

SUMMARY

The invention overcomes the identified problems by providing a method for analyzing large disk resident data sets. In a computer system having a large disk-resident data set, a 5 method of analyzing the disk-resident data set using the Patient Rule Induction Method (PRIM) is disclosed wherein a relational data table is initially received. The relational data table includes continuous attributes, discrete attributes, a meta parameter and a cost attribute. The cost attribute represents cost output values based on continuous attribute values and discrete attribute values as inputs. A hyper-rectangle is then formed which encloses a multi- 10 dimensional space defined by the continuous attribute values and the discrete attribute values. The continuous attribute values and the discrete attribute values are represented as points within the multi-dimensional space. Once the hyper-rectangle is formed, PRIM is completed with a peeling phase and a pasting phase. A plurality of points along edges of the hyper-rectangle are then removed (peeled) based on an average of the cost output value from the plurality of points until a count of the points enclosed within the hyper-rectangle equals the meta parameter. Discrete attribute values and continuous attribute values which were 15 removed from the hyper-rectangle are next added (~~peeled~~^{pasted}) along edges of the hyper-rectangle until a sum of the cost output value over the multi-dimensional space enclosed by the hyper-rectangle changes. In a further embodiment of the invention utilizing a parallel architecture computer system having a plurality of processors, calculations of the cost attribute average 20 values over the plurality of points enclosed by the hyper-rectangle are performed in parallel across the plurality of processors using reduction and a one to all broadcast.

The invention provides many advantages over known techniques. The present 25 invention includes the ability to analyze large disk resident data sets without having to load the data set into main memory. In addition, the invention can be practiced on a parallel computer architecture or a symmetric multi-processor architecture to improve performance. Moreover, the memory requirements of the invention are only limited by the size of the cost attribute sought to be maximize or minimized.

30

BRIEF DESCRIPTION OF THE DRAWINGS

Additional advantages and features of the invention will become readily apparent upon reading the following detailed description and appended claims when taken in conjunction with reference to the drawings, in which:

Figure 1 is a plot depicting the hyper-rectangles determined using the methods of the present invention as compared to the prior art SLIQ invention;

Figure 2 depicts data structures used by the Patient Rule Induction Method for analysis of small data sets;

Figure 3 depicts a computer system according to an embodiment of the invention;

Figure 4 is a flow chart depicting operation of the invention according to an embodiment of the invention;

Figure 5 depicts additional method steps for defining the hyper-rectangle according to a further embodiment of the invention;

Figure 6 depicts additional method steps for peeling procedures according to a further embodiment of the invention;

Figure 7 depicts additional method steps for optimizing calculation of the lowest average cost output value according to a further embodiment of the invention;

Figure 8 depicts additional method steps for the peeling procedures according to a further embodiment of the invention;

Figure 9 depicts additional method steps for performing the pasting procedures according to a further embodiment of the invention;

Figure 10 depicts additional method steps for optimizing calculation of the highest average cost output value according to a further embodiment of the invention;

5 Figure 11 depicts additional method steps for the pasting procedures according to a further embodiment of the invention;

Figure 12 depicts a parallel computer architecture according to an embodiment of the invention;

10 Figure 13 is a flow chart depicting operation of the Patient Rule Induction Method in a parallel computer architecture;

Figure 14 depicts data redistribution among the plurality of processors using striping according to an embodiment of the invention;

15 Figure 15 depicts additional method steps for performing the peeling procedures according to a further embodiment of the invention;

20 Figure 16 depicts additional method steps for performing the pasting procedures according to a further embodiment of the invention;

Figure 17 depicts a symmetric multi-processor computer system according to an embodiment of the invention;

25 Figure 18 is a flow chart depicting operation of the Patient Rule Induction Method on a symmetric multi-processor computer architecture;

Figure 19 depicts additional method steps for performing the peeling procedures according to a further embodiment of the invention; and

30

Figure 20 depicts additional method steps for performing the pasting procedures according to a further embodiment of the invention.

DETAILED DESCRIPTION

5

The present invention relates generally to data analysis involving the prediction of high values of some output variable based on a set of input variables. In particular, the invention relates to analysis of large, disk resident data sets using the Patient Rule Induction Method (PRIM). The PRIM method for large disk resident data sets includes two phases for 10 generating a hyper-rectangle. These phases are the Peeling phase and the Pasting phase. We assume that the data is input as a relational data table. The relational data table contains A_c continuous attributes, A_d discrete attributes and a cost attribute or output value based on the A_c continuous attributes and A_d discrete attributes as input. In order to simplify this discussion, the term cost attribute is generally used to refer to output values based on continuous attribute values and discrete attribute values as inputs. In other words, a cost output value is an output based on a continuous attribute value and a discrete attribute value as inputs. β_0 is meta parameter given by the user. It represents the minimum number of points present in the hyper-rectangle.

20 An exemplary embodiment of the invention is described with reference to Figures 3 and 4. Figure 3 depicts a computer system 10 according to an embodiment of the invention. The computer system includes a processor (CPU) 12 coupled to a bus 14. A random access memory 16 and a hard disk memory 18 are also coupled to the bus 14 and are accessible by the processor. The hard disk 18 is configured to store programs and data necessary for the invention, as described below. An optional user interface 20 is provided for input of raw 25 data. In one embodiment, the input/output devices include a keyboard, mouse, and monitor. An optional network interface 21 is also provided. It should be appreciated that raw data may be entered in various different ways, for example via the user interface 20 or the network interface 21.

30

The hard disk 18 is configured to store the program and data in the computer system
10. The memory, including the RAM and the hard disk, is divided into three primary
components: communications procedures 22, control procedures 28, and data 42. The
communications procedures include routines 24 and 26 for identifying large disk resident data
5 sets. The control procedures include routines 30-46 that perform data analysis functions
according to the invention. The data portion creates indexes to an A_c continuous attribute
lists and a discrete attribute list of the relational data table 54, and stores a cost list, and a
meta parameter 50 in main memory (RAM) 16. These routines are described in greater detail
below.

10 Operation of the invention is described with reference to the Figure 4 flowchart. At
step 102, a meta parameter and a relational data table are received. As described above, the
relational data table includes A_c continuous attributes, discrete attributes and a cost attribute.
The cost attribute represents cost output values based on continuous attributes values and
discrete attribute values as inputs. The attributes are originally rows or tuples that form the
relational data table which also contains an index to identify the tuple when the attributes are
separated. At step 110, a hyper-rectangle is defined by enclosing a multi-dimensional space
defined by the continuous attribute values and the discrete attribute values. To form the
multi-dimensional space, the continuous attribute values and the discrete attribute values are
represented as points within the multi-dimensional space. At steps 130 and 200, a plurality of
points along edges of the hyper-rectangle are removed (peeled) based on an average of the
cost output value from the plurality of points. The points are repeatedly removed until a
count of the points enclosed within the hyper-rectangle equals the meta parameter. Finally at
steps 290 and 300, removed discrete attribute value points and the continuous attribute value
points are added (pasted) along the edges of the hyper-rectangle until a sum of the cost output
values over the multi-dimensional space enclosed by the hyper-rectangle changes.

Figure 5 depicts additional methods steps for defining the hyper-rectangle of step 110
according to a further embodiment of the invention. Figure 5 depicts vertical partitioning of
30 the relational data table that enables analysis of the large disk resident data set without having
to load the large disk resident data set into main memory 16 as required by conventional

systems. The data is separated into $A_c + 2$ lists. At step 112, a list is generated for each continuous attribute to form A_c continuous attribute lists containing the continuous attribute values. At step 114, a discrete attribute list containing the A_d discrete attributes and the discrete attribute values is generated. At step 116, a cost attribute list is generated containing the cost output values and is stored in main memory. At step 118, a label is added to each of the A_c continuous attribute lists, the discrete attribute list and the cost attribute list. The label is an index of the tuple to which the respective attribute value belongs within the relational data table. The index allows the continuous attribute values and discrete attribute values to reference their respective tuple and cost output value.

10

At step 120, a label is added to the cost list. The label is a cost flag that indicates whether the tuple containing the cost output value is enclosed within the hyper-rectangle. The cost flag is initially set to one indicating the tuple is enclosed within the hyper-rectangle, since all points are initially enclosed within the hyper-rectangle. Finally at step 122, a further label is added to the cost list. The label is a cost counter of the number of points of the tuple that are not enclosed within the hyper-rectangle. The cost counter is initially set to one indicating since all points are initially enclosed with the hyper-rectangle. The cost counter is used in the pasting procedures 42.

15
20
25

Figure 6 depicts additional methods steps for the peeling procedures 36 of step 200 according to a further embodiment of the invention. At step 210, a discrete attribute value enclosed within the plurality of points along an edge of the hyper-rectangle with a lowest average cost output value is found. At step 220, the continuous attribute value enclosed within the plurality of points along an edge of the hyper-rectangle with a lowest average cost output value is found. At step 240, the lowest average cost output value determined in step 210 is compared with the lowest average cost output value determined in step 220 to determine an attribute with the lowest average cost output value. At step 250, all continuous attribute value points and all discrete attribute value points of the tuples containing the attribute determined in step 240 are removed (peeled) from the hyper-rectangle. At step 270, steps 210 to 250 are repeated until the count of the points within the hyper-rectangle equals β_0 .

Finally at step 280, a total cost output is calculated. The total cost output is a sum of the cost output values over the multi-dimensional space enclosed by the hyper-rectangle. Although the peeling procedure described above is maximizing the points enclosed by the hyper-rectangle, the points enclosed by the hyper-rectangle could be minimized as well. In 5 addition, the discrete attribute value of step 200 is, for example, determined using a plurality of discrete histogram for each discrete attribute. Each histogram contains the respective distinct discrete attribute values and an average of the cost output value for each tuple containing the discrete attribute value. The average cost output value is calculated by using the index of the discrete attribute value and totaling each cost output value in the cost list 10 with a matching index and the cost flag equal to zero. The total is then divided by a count of the cost output values in the total to form the average cost output value for the respective discrete attribute value. The discrete attribute value with the lowest average cost output value is then easily chosen by examining each discrete histogram.

15 Figure 7 depicts additional methods steps for optimizing calculation of the lowest average cost output value of step 220 according to a further embodiment of the invention. Initially, each continuous attribute list is sorted in increasing order with a start pointer to a first row in each continuous attribute list. At step 222, a second continuous attribute list is generated for each of the A_c continuous attribute lists. The second continuous attribute lists are sorted in decreasing order with an end pointer to a first row in each second continuous attribute list. At step 224, a start cutoff value is marked in each of the A_c continuous attribute lists based on a count of the tuples containing the discrete attribute value determined in step 20 210 and enclosed within the hyper-rectangle. At step 226, an end cutoff value is marked in each of the second continuous attribute lists based on the count of the tuples containing the discrete attribute value determined in step 210 and enclosed within the hyper-rectangle.

25 At step 228, a start cost average value is determined for each continuous output value between the start pointer and the cutoff value for each of the A_c cost attribute lists. At step 230, an end cost average value is determined for each continuous output value between the cutoff value and the end pointer for each of the second cost attribute lists. At step 232, A_c continuous histograms, each containing the continuous attribute and the average cost output

value, are generated. The average cost output value is a lesser of the start cost average value and the end cost average value. Finally at step 234, the average cost output value for each continuous histogram is compared to determine the continuous attribute value with the lowest average cost output value.

5

Figure 8 depicts additional methods steps for the peeling procedures 36 of step 250 according to a further embodiment of the invention. In order to optimize this process, the discrete attribute list is sorted based on an assigned code thereby grouping the discrete attribute values according to their discrete attribute. At step 252, when the attribute determined in step 240 is a continuous attribute and the start cost average value is less than the end cost average value (step 254), the cost flag is set to zero for each continuous attribute value between the start pointer and the start cutoff value at step 256. The index of the continuous attribute value is used to reference and modify the cost flag. At step 258, the start pointer is set equal to the start cutoff value. At step 254, when the end cost average value is less than the start cost average value, the cost flag is set to zero for each continuous attribute value between the end cutoff value and the end pointer at step 260. The index of the continuous attribute value is used to reference and modify the cost flag. At step 258, the end pointer is set equal to the end cutoff value. At step 252, when the attribute determined in step 240 is a discrete attribute, the cost flag is set to zero for each discrete attribute value equal to the attribute determined in step 240. The index of the discrete attribute value is used to reference and modify the cost flag.

Figure 9 depicts additional methods steps for performing the pasting procedures 42 of step 300 according to a further embodiment of the invention. At step 310, for each tuple with the cost flag set to zero, the cost counter is incremented for each point belonging to the tuple that is not enclosed with the hyper-rectangle. The cost counter is used to determine whether a point is inside or outside the hyper-rectangle. At step 320, the discrete attribute value outside of the points enclosed by the hyper-rectangle with a highest average cost output value is determined. At step 330, the continuous attribute value outside of the points enclosed by the hyper-rectangle with a highest average cost output value is determined. At step 350, the highest average cost output value determined in step 320 is compared with the highest

average cost output value determined in step 330 to find an attribute with the highest average cost output value. Finally at step 360, the cost counter for all continuous attribute value points and all discrete attribute value points belonging to the tuples containing the attribute determined in step 350 are decremented. Attributes with the cost counter equal to zero are
5 enclosed within the hyper-rectangle, otherwise the attribute is outside the hyper-rectangle. In other words, incrementing of the cost counter removes attributes from the hyper-rectangle, while decrementing of the cost counter to equal zero adds attributes to the hyper-rectangle. At step 370, steps 310 to 360 are repeated until a sum of the cost output values over the plurality of points enclosed by the hyper-rectangle is less than the total cost output. This
10 pasting procedure ensures a more thorough representation of high value points. As described above, the highest average cost output value of step 320 is calculated using a plurality of discrete histograms.

Figure 10 depicts additional methods steps for optimizing calculation of the highest average cost output value of step 330 according to a further embodiment of the invention. At step 332, the continuous attribute values between a first row of the A_c cost attribute lists and the start pointer are sorted in decreasing order. At step 334, a start cutoff value is marked in each of the A_c continuous attribute lists based on a count of the tuples containing the discrete attribute value determined in step 320 and enclosed within the hyper-rectangle. At step 336, an end cutoff value is marked in each of the second continuous attribute lists based on the count of the tuples containing the discrete attribute value determined in step 320 and enclosed within the hyper-rectangle.

At step 338, a start cost average value is determined for each continuous output value between the start pointer and the cutoff value for each of the A_c cost attribute lists. At step 340, an end cost average value is determined for each continuous output value between the cutoff value and the end pointer for each of the second cost attribute lists. At step 342, A_c continuous histograms, each containing the continuous attribute and the average cost output value, are generated. The average cost output value is a greater of the start cost average value
25 and the end cost average value. Finally at step 344, the average cost output value for each
30

continuous histogram is compared to determine the continuous attribute value with the highest average cost output value.

Figure 11 depicts additional methods steps for the pasting procedures 42 of step 360 according to a further embodiment of the invention. In order to optimize this process, the discrete attribute list is sorted based on an assigned code thereby grouping the discrete attribute values according to their discrete attribute. At step 362, when the attribute determined in step 350 is a continuous attribute and the start cost average value is less than the end cost average value (step 364), decrement the cost counter for each continuous attribute value between the start pointer and the start cutoff value at step 366. The index of the continuous attribute value is used to reference and modify the cost counter. At step 368, the start pointer is set equal to the start cutoff value. At step 374, when the end cost average value is less than the start cost average value, decrement the cost counter for each continuous attribute value between the end cutoff value and the end pointer at step 370. The index of the continuous attribute value is used to reference and modify the cost counter. At step 372, the end pointer is set equal to the end cutoff value. At step 362, when the attribute determined in step 350 is a discrete attribute, decrement the cost counter for each discrete attribute value equal to the attribute determined in step 350. The index of the discrete attribute value is used to reference and modify the cost counter.

A further embodiment of the invention is described with reference to Figures 12 and 13. Figure 12 depicts a parallel architecture computer system 400 according to an embodiment of the invention. The computer system includes a plurality of processors 412A-412N coupled to a bus 414. A random access memory 416 for each processor 412 and a hard disk memory 418 are also coupled to the bus 414 and are accessible by the plurality of processors 412. The hard disk 418 is configured to store programs and data necessary for the invention, as described below. An optional user interface 420 is provided for input of raw data. In one embodiment, the input/output devices include a keyboard, mouse, and monitor. An optional network interface 421 is also provided. It should be appreciated that raw data may be entered in various different ways, for example via the user interface 18 or the network interface 20.

The hard disk 418 is configured to store the program and data in the computer system
10. The memory, including the RAM and the hard disk, is divided into three primary
components: communications procedures 422, control procedures 432, and data 452. The
communications procedures include routines 424 - 430 for identifying large disk resident data
5 sets and providing communication among the plurality of processors. The control procedures
include routines 434-450 that perform the invention's data analysis functions. The data
portion creates indexes to an A_c continuous attribute lists and a discrete attribute list of the
relational data table 454, and stores a cost list, and a meta parameter 456 in main memory
(RAM) 416. These routines are described in greater detail below.

10

Operation of the invention is described with reference to the Figure 13 flowchart. At
step 502, a meta parameter and a relational data table are received. As described above, the
relational data table is includes A_c continuous attributes, discrete attributes and a cost
attribute. The cost attribute represents cost output values based on continuous attributes
values and discrete attribute values as inputs. At step 504, a hyper-rectangle is defined by
enclosing a multi-dimensional space defined by the continuous attribute values and the
discrete attribute values. At step 506, the data is separated into $A_c + 2$ lists. A list is
generated for each continuous attribute to form A_c continuous attribute lists containing the
continuous attribute values A discrete attribute list containing the A_d discrete attributes and
the discrete attribute values. A cost attribute list containing the cost output values is stored in
main memory. Each of the A_c continuous attribute lists, the discrete attribute list and the cost
attribute list also include a label. The label is an index of the tuple to which the respective
attribute value belongs within the relational data table. The index allows the continuous
attribute values and discrete attribute values to reference their respective tuple and cost output
value. The lists also include a cost counter label. The cost counter is a count of the number
20 of points of the tuple that are not enclosed within the hyper-rectangle as described above.
25

25

At step 508, the A_c continuous attribute lists are sorted in parallel among the plurality
of processors based on the continuous attribute values in each row of the cost attribute lists.
30 At step 510, the attribute lists are stripe across the plurality of processors as depicted in
Figure 14. At steps 512 and 520, a plurality of points along edges of the hyper-rectangle are

removed (peeled) based on an average of the cost output value from the plurality of points using reduction and a one to all broadcast with the plurality of processors. The points are repeatedly removed until a count of the points enclosed within the hyper-rectangle equals the meta parameter. Finally at steps 550 and 600, removed discrete attribute value points and the 5 continuous attribute value points are added (pasted) along the edges of the hyper-rectangle until a sum of the cost output values over the multi-dimensional space enclosed by the hyper-rectangle changes using reduction and a one to all broadcast with the plurality of processors.

Some synchronization steps may be required for getting the exact fraction of points 10 for the continuous attributes. This is due to possibility of some continuous attribute values in an attribute list no longer being present inside the hyper-rectangle. Details concerning the communication and synchronization among the plurality of processors will be apparent to those skilled in the art of parallel computer architectures and are therefore not set forth herein except as needed for a proper understanding of the invention. However, a reduction step for all the processors can be performed to get the average value for a particular attribute during the peeling phase. The best attribute can be selected for peeling. After the peeling has been performed the peeled transaction id's are transferred to each processor so that it can update the tags for its memory resident cost attribute list. Conventional reduction, implemented as a converse to the one to all broadcast, is well known in the field of parallel computing.

(*Introduction to Parallel Computing: Design and Analysis of Algorithms* by Vipin Kumar, 15 Ananth Grama, Anshul Gupta, and George Karypis Chapter 3, § 3.2.)

Figure 15 depicts additional methods steps for the peeling procedures 440 of step 520 according to a further embodiment of the invention. At step 522, a discrete attribute value 20 enclosed within the plurality of points along an edge of the hyper-rectangle with a lowest average cost output value is found using reduction among the processors. At step 524, the continuous attribute value enclosed within the plurality of points along an edge of the hyper-rectangle with a lowest average cost output value is found using reduction. At step 526, the lowest average cost output value determined in step 522 is compared with the lowest average 25 cost output value determined in step 524 to determine an attribute with the lowest average cost output value. At step 528, all continuous attribute value points and all discrete attribute 30 cost output value points are removed.

value points of the tuples containing the attribute determined in step 526 are removed (peeled) from the hyper-rectangle using a one to all broadcast between the processors. At step 530, steps 522 to 528 are repeated until the count of the points within the hyper-rectangle equals β_0 .

5

Finally at step 532, a total cost output is calculated. The total cost output is a sum of the cost output values over the multi-dimensional space enclosed by the hyper-rectangle. Although the peeling procedure described above is maximizing the points enclosed by the hyper-rectangle, the points enclosed by the hyper-rectangle could be minimized as well. As 10 describe above, the lowest average cost output value of step 520 can be calculated using a plurality of discrete histograms.

300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990
1000
1010
1020
1030
1040
1050
1060
1070
1080
1090
1100
1110
1120
1130
1140
1150
1160
1170
1180
1190
1200
1210
1220
1230
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
2980
2990
3000
3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010
4020
4030
4040
4050
4060
4070
4080
4090
4100
4110
4120
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500
4510
4520
4530
4540
4550
4560
4570
4580
4590
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4910
4920
4930
4940
4950
4960
4970
4980
4990
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140
5150
5160
5170
5180
5190
5200
5210
5220
5230
5240
5250
5260
5270
5280
5290
5300
5310
5320
5330
5340
5350
5360
5370
5380
5390
5400
5410
5420
5430
5440
5450
5460
5470
5480
5490
5500
5510
5520
5530
5540
5550
5560
5570
5580
5590
5600
5610
5620
5630
5640
5650
5660
5670
5680
5690
5700
5710
5720
5730
5740
5750
5760
5770
5780
5790
5800
5810
5820
5830
5840
5850
5860
5870
5880
5890
5900
5910
5920
5930
5940
5950
5960
5970
5980
5990
6000
6010
6020
6030
6040
6050
6060
6070
6080
6090
6100
6110
6120
6130
6140
6150
6160
6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340
6350
6360
6370
6380
6390
6400
6410
6420
6430
6440
6450
6460
6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050
7060
7070
7080
7090
7100
7110
7120
7130
7140
7150
7160
7170
7180
7190
7200
7210
7220
7230
7240
7250
7260
7270
7280
7290
7300
7310
7320
7330
7340
7350
7360
7370
7380
7390
7400
7410
7420
7430
7440
7450
7460
7470
7480
7490
7500
7510
7520
7530
7540
7550
7560
7570
7580
7590
7600
7610
7620
7630
7640
7650
7660
7670
7680
7690
7700
7710
7720
7730
7740
7750
7760
7770
7780
7790
7800
7810
7820
7830
7840
7850
7860
7870
7880
7890
7900
7910
7920
7930
7940
7950
7960
7970
7980
7990
8000
8010
8020
8030
8040
8050
8060
8070
8080
8090
8100
8110
8120
8130
8140
8150
8160
8170
8180
8190
8200
8210
8220
8230
8240
8250
8260
8270
8280
8290
8300
8310
8320
8330
8340
8350
8360
8370
8380
8390
8400
8410
8420
8430
8440
8450
8460
8470
8480
8490
8500
8510
8520
8530
8540
8550
8560
8570
8580
8590
8600
8610
8620
8630
8640
8650
8660
8670
8680
8690
8700
8710
8720
8730
8740
8750
8760
8770
8780
8790
8800
8810
8820
8830
8840
8850
8860
8870
8880
8890
8900
8910
8920
8930
8940
8950
8960
8970
8980
8990
9000
9010
9020
9030
9040
9050
9060
9070
9080
9090
9100
9110
9120
9130
9140
9150
9160
9170
9180
9190
9200
9210
9220
9230
9240
9250
9260
9270
9280
9290
9300
9310
9320
9330
9340
9350
9360
9370
9380
9390
9400
9410
9420
9430
9440
9450
9460
9470
9480
9490
9500
9510
9520
9530
9540
9550
9560
9570
9580
9590
9600
9610
9620
9630
9640
9650
9660
9670
9680
9690
9700
9710
9720
9730
9740
9750
9760
9770
9780
9790
9800
9810
9820
9830
9840
9850
9860
9870
9880
9890
9900
9910
9920
9930
9940
9950
9960
9970
9980
9990
10000
10010
10020
10030
10040
10050
10060
10070
10080
10090
100100
100110
100120
100130
100140
100150
100160
100170
100180
100190
100200
100210
100220
100230
100240
100250
100260
100270
100280
100290
100300
100310
100320
100330
100340
100350
100360
100370
100380
100390
100400
100410
100420
100430
100440
100450
100460
100470
100480
100490
100500
100510
100520
100530
100540
100550
100560
100570
100580
100590
100600
100610
100620
100630
100640
100650
100660
100670
100680
100690
100700
100710
100720
100730
100740
100750
100760
100770
100780
100790
100800
100810
100820
100830
100840
100850
100860
100870
100880
100890
100900
100910
100920
100930
100940
100950
100960
100970
100980
100990
1001000
1001010
1001020
1001030
1001040
1001050
1001060
1001070
1001080
1001090
1001100
1001110
1001120
1001130
1001140
1001150
1001160
1001170
1001180
1001190
1001200
1001210
1001220
1001230
1001240
1001250
1001260
1001270
1001280
1001290
1001300
1001310
1001320
1001330
1001340
1001350
1001360
1001370
1001380
1001390
1001400
1001410
1001420
1001430
1001440
1001450
1001460
1001470
1001480
1001490
1001500
1001510
1001520
1001530
1001540
1001550
1001560
1001570
1001580
1001590
1001600
1001610
1001620
1001630
1001640
1001650
1001660
1001670
1001680
1001690
1001700
1001710
1001720
1001730
1001740
1001750
1001760
1001770
1001780
1001790
1001800
1001810
1001820
1001830
1001840
1001850
1001860
1001870
1001880
1001890
1001900
1001910
1001920
1001930
1001940
1001950
1001960
1001970
1001980
1001990
1002000
1002010
1002020
1002030
1002040
1002050
1002060
1002070
1002080
1002090
1002100
1002110
1002120
1002130
1002140
1002150
1002160
1002170
1002180
1002190
1002200
1002210
1002220
1002230
1002240
1002250
1002260
1002270
1002280
1002290
1002300
1002310
1002320
1002330
1002340
1002350
1002360
1002370
1002380
1002390
1002400
1002410
1002420
1002430
1002440
1002450
1002460
1002470
1002480
1002490
1002500
1002510
1002520
1002530
1002540
1002550
1002560
1002570
1002580
1002590
1002600
1002610
1002620
1002630
1002640
1002650
1002660
1002670
1002680
1002690
1002700
1002710
1002720
1002730
1002740
1002750
1002760
1002770
1002780
1002790
1002800
1002810
1002820
1002830
1002840
1002850
1002860
1002870
1002880
1002890
1002900
1002910
1002920
1002930
1002940
1002950
1002960
1002970
1002980
1002990
1003000
1003010
1003020
1003030
1003040
1003050
1003060
1003070
1003080
1003090
1003100
1003110
1003120
1003130
1003140
1003150
1003160
1003170
1003180
1003190
1003200
1003210
1003220
1003230
1003240
1003250
1003260
1003270
1003280
1003290
1003300
1003310
1003320
1003330
1003340
1003350
1003360
1003370
1003380
1003390
1003400
1003410
1003420
1003430
1003440
1003450
1003460
1003470
1003480
1003490
1003500
1003510
1003520
1003530
1003540
1003550
1003560
1003570
1003580
1003590
1003600
1003610
1003620
1003630
1003640
1003650
1003660
1003670
1003680
1003690
1003700
1003710
1003720
1003730
1003740
1003750
1003760
1003770
1003780
1003790
1003800
1003810
1003820
1003830
1003840
1003850
1003860
1003870
1003880
1003890
1003900
1003910
1003920
1003930
1003940
1003950
1003960
1003970
1003980
1003990
1004000
1004010
1004020
1004030
1004040
1004050
1004060
1004070
1004080
1004090
1004100
1004110
1004120
1004130
1004140
1004150
1004160
1004170
1004180
1004190
1004200
1004210
1004220
1004230
1004240
1004250
1004260
1004270
1004280
1004290
1004300
1004310
1004320
1004330
1004340
1004350
1004360
1004370
1004380
1004390
1004400
1004410
1004420
1004430
1004440
1004450
1004460
1004470
1004480
1004490
1004500
1004510
1004520
1004530
1004540
1004550
1004560
1004570
1004580
1004590
1004600
1004610
1004620
1004630
1004640
1004650
1004660
1004670
1004680
1004690
1004700
1004710
1004720
1004730
1004740
1004750
1004760
1004770
1004780
1004790
1004800
1004810
1004820
1004830
1004840
1004850
1004860
1004870
1004880
1004890
1004900
1004910
1004920
1004930
1004940
1004950
1004960
1004970
1004980
1004990
1005000
1005010
1005020
1005030
1005040
1005050
1005060
1005070
1005080
1005090
1005100
1005110
1005120
1005130
1005140
1005150
1005160
1005170
1005180
1005190
1005200
1005210
1005220
1005230
1005240
1005250
1005260
1005270
1005280
1005290
1005300
1005310
1005320
1005330
1005340
1005350
1005360
1005370
1005380
1005390
1005400
1005410
1005420
1005430
1005440
1005450
1005460
1005470
1005480
1005490
1005500
1005510
1005520
1005530
1005540
1005550
1005560
1005570
1005580
1005590
1005600
1005610
1005620
1005630
1005640
1005650
1005660
1005670
1005680
1005690
1005700
1005710
1005720
1005730
1005740
1005750
1005760
1005770
1005780
1005790
1005800
1005810
1005820
1005830
1005840
1005850
1005860
1005870
1005880
1005890
1005900
1005910
1005920
1005930
1005940
1005950
1005960
1005970
1005980
1005990
1006000
1006010
1006020
1006030
1006040
1006050
1

A further embodiment of the invention is described with reference to Figures 17 and 18. Figure 17 depicts a symmetric multi-processor computer system 700 according to an embodiment of the invention. The symmetric multi-processor (SMP) 712 coupled to a bus 714. A random access memory 716 and a share hard disk memory 717 are also coupled to the bus 714. The shared disk 717 is partitioned into a plurality of portion that are accessible by the symmetric multi-processors 712. The main memory (RAM) 716 is configured to store programs and data necessary for the invention, as described below. An optional user interface 718 is provided for input of raw data. In one embodiment, the input/output devices include a keyboard, mouse, and monitor. An optional network interface 721 is also provided. It should be appreciated that raw data may be entered in various different ways, for example via the user interface 720 or the network interface 721.

The hard disk 718 is configured to store the program and data in the computer system 700. The memory, including the RAM and the hard disk, is divided into three primary components: communications procedures 722, control procedures 732, and data 752. The communications procedures include routines 724 - 730 for identifying large disk resident data sets and providing communication for the symmetric multi-processors 712. The control procedures include routines 735-754 that perform the invention's data analysis functions. The data portion creates indexes to an A_c continuous attribute lists and a discrete attribute list of the relational data table 754, and stores a cost list, and a meta parameter 762 in main memory (RAM) 716. These routines are described in greater detail below.

Operation of the invention is described with reference to the Figure 18 flowchart. At step 802, a meta parameter and a relational data table are received. As describe above, the relational data table is includes A_c continuous attributes, discrete attributes and a cost attribute. The cost attribute represents cost output values based on continuous attributes values and discrete attribute values as inputs. At step 804, a hyper-rectangle is defined by enclosing a multi-dimensional space defined by the continuous attribute values and the discrete attribute values. At step 806, the data is separated into $A_c + 2$ lists. A list is generated for each continuous attribute to form A_c continuous attribute lists containing the continuous attribute values A discrete attribute list containing the A_d discrete attributes and

the discrete attribute values. A cost attribute list containing the cost output values is stored in main memory. Each of the A_c continuous attribute lists, the discrete attribute list and the cost attribute list also include a label. The label is an index of the tuple to which the respective attribute value belongs within the relational data table. The index allows the continuous attribute values and discrete attribute values to reference their respective tuple and cost output value. The lists also include a cost counter label. The cost counter is a count of the number of points of the tuple that are not enclosed within the hyper-rectangle as described above.

At step 808, the A_c continuous attribute lists are sorted in parallel among the plurality of shared disk portions based on the continuous attribute values in each row of the cost attribute lists. At step 810, the attribute lists are stripe across the plurality of shared disk portions as depicted in Figure 14. At steps 812 and 820, a plurality of points along edges of the hyper-rectangle are removed (peeled) based on an average of the cost output value from the plurality of points using reduction and a one to all broadcast within the symmetric multi-processors. The points are repeatedly removed until a count of the points enclosed within the hyper-rectangle equals the meta parameter. Finally at steps 850 and 900, removed discrete attribute value points and the continuous attribute value points are added (pasted) along the edges of the hyper-rectangle until a sum of the cost output values over the multi-dimensional space enclosed by the hyper-rectangle changes using reduction and a one to all broadcast within the symmetric multi-processors.

Figure 19 depicts additional methods steps for the peeling procedures 740 of step 820 according to a further embodiment of the invention. At step 822, a discrete attribute value enclosed within the plurality of points along an edge of the hyper-rectangle with a lowest average cost output value is found using reduction within the symmetric multi-processor. At step 824, the continuous attribute value enclosed within the plurality of points along an edge of the hyper-rectangle with a lowest average cost output value is found using reduction. At step 826, the lowest average cost output value determined in step 822 is compared with the lowest average cost output value determined in step 824 to determine an attribute with the lowest average cost output value. At step 828, all continuous attribute value points and all discrete attribute value points of the tuples containing the attribute determined in step 826 are

removed (peeled) from the hyper-rectangle using a one to all broadcast within the symmetric multi-processors. At step 830, steps 822 to 828 are repeated until the count of the points within the hyper-rectangle equals β_0 . Finally at step 832, the total cost output is calculated as described above. As described above, the lowest average cost output value of step 520 can be

5 calculated using a plurality of discrete histograms.

Figure 20 depicts additional methods steps for performing the pasting procedures 746 of step 900 according to a further embodiment of the invention. At step 902, for each tuple with the cost flag set to zero, incrementing the cost counter for each point belonging to the tuple that is not enclosed with the hyper-rectangle using a one to all broadcast within the symmetric multi-processors. At step 904, the discrete attribute value outside of the points enclosed by the hyper-rectangle with a highest average cost output value is determined using reduction and a one to all broadcast within the symmetric multi-processors. At step 906, the continuous attribute value outside of the points enclosed by the hyper-rectangle with a highest average cost output value is determined using reduction and a one to all broadcast within the symmetric multi-processors. At step 908, the highest average cost output value determined in step 904 is compared with the highest average cost output value determined in step 906 to find an attribute with the highest average cost output value. Finally at step 910, the cost counter for all continuous attribute value points and all discrete attribute value points belonging to the tuples containing the attribute determined in step 908 are decremented using a one to all broadcast within the symmetric multi-processors. Attributes with the cost counter equal to zero are enclosed within the hyper-rectangle. At step 912, steps 902 to 910 are repeated until a sum of the cost output value over the plurality of points enclosed by the hyper-rectangle is less than the total cost output. This pasting procedure ensure a more through representation of high value points. As described above, the highest average cost output value of step 904 is calculated using a plurality of discrete histograms.

Exemplary embodiments have been described with reference to specific configurations. Those skilled in the art will appreciate that various changes and modifications can be made while remaining within the scope of the claims. It is to be understood that even though numerous characteristics and advantages of various

embodiments of the present invention have been set forth in the foregoing description, together with details of the structure and function of various embodiments of the invention, this disclosure is illustrative only, and changes may be made in detail, especially in matters of structure and arrangement of parts within the principles of the present invention to the full extent indicated by the broad general meaning of the terms in which the appended claims are expressed.

For example, it is within the contemplation of the present invention to practice the invention with a conventional computer without departing from the scope of the invention. Moreover, teaching of the present invention are not limited to maximization of some output value over a multi-dimensional space of input values. The invention may be practiced by minimizing the points over the multi-dimensional space without departing from the scope of the invention.

The invention provides many advantages over known techniques. The present invention includes the ability to analyze large, disk resident data sets without having to load the data set into main memory. In addition, the invention can be practiced on a parallel computer architecture or a symmetric multi-processor architecture to improve performance. Moreover, the memory requirements of the invention are only limited by the size of the cost attribute sought to be maximized or minimized.

Having disclosed exemplary embodiments and the best mode, modifications and variations may be made to the disclosed embodiments while remaining within the scope of the invention as defined by the following claims.